UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/447,501	11/23/1999	LANDY WANG	2260	3903

7590 02/25/2004

LAW OFFICES OF ALBERT S. MICHALIK, PLLC
704 - 228TH AVENUE NE.
SUITE 193
SAMMAMISH, WA 98074

EXAMINER	
ANYA, CHARLES E	

ART UNIT	PAPER NUMBER
2126	14

DATE MAILED: 02/25/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/447,501	WANG ET AL.	
	Examiner	Art Unit	
	Charles E Anya	2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3/ MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-16 and 27-53* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-16 and 27-53* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-16 and 27-53 are pending in this application.

Specification

2. Please furnish the office with the references (Inside Windows NT by Helen Custer, Microsoft Press 1993 and Inside Windows NT Second Edition by David A. Solomon Microsoft Press 1998) cited on page 19 lines 1 – 15.

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.
The following title is suggested: Method and system for monitoring and testing software drivers memory allocation request.

4. Applicant's abstract is longer than 150 words, as a result requires amendment.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 31 and 35-37 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Pat. No. 5,491,808 to Geist, Jr.

7. As to claim 31, Geist teaches a computer-readable medium having computer executable instructions, comprising: receiving a plurality of requests from a component driver for allocation of various distinct sets of memory ("...allocation calls..." Col. 7 Ln. 14 – 29), allocating the memory, memory allocated to the component driver on each request (Step 5 Col. 8 Ln. 26 – 35), tracking the receiving requests for de-allocation of at least one of the sets of memory allocated to the driver/tracking the space memory de-allocated in each received de-allocation request ("...removing..." Col. 10 Ln. 45 – 63), determining from the tracking whether memory remains allocated to the driver at a time when the driver should have no space memory allocated thereto and generating, an error at the time if memory remains allocated ("...unfreed memory..." Col. 6 Ln. 9 – 16).

8. As to claim 35, Geist teaches a system for monitoring drivers, comprising: an operating system component including an interface for receiving requests from drivers; a re-vectoring component for examining the requests to determine whether requesting drivers are to be monitored ((JMP) Col. 7 Ln. 42 – 50, Col. 8 Ln. 10 – 25, Step 100 Col. 10 Ln. 1 – 7), and a driver verifier component operably connected to the re-vectoring component, the driver verifier receiving information from the re-vectoring component for a driver that is to be monitored ("...thunk..." Col. 7 Ln. 42 – 62) and executing at least one test to monitor the driver, wherein in response to a request for memory from the

driver, the driver verifier component allocates memory for the driver to use from a pool of memory other than a memory pool normally allocated from when the driver is operating unmonitored (ABLK...MSG free pool..." Col. 8 Ln. 27 – 35).

9. As to claim 36, Geist teaches the system of claim 35 wherein the operating system component comprises a kernel component (...kernel..." Col. 9 Ln. 19 – 27).

10. As to claim 37, Geist teaches the system of claim 35 wherein the re-vectoring component determines that the driver is to be monitored based on a setting in a registry ("...list..." Col. 9 Ln. 23 – 27).

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. **Claims 1-9,11-16,27-30,32-34,39-53 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,491,808 to Geist, Jr. in view of U.S. Pat. No. 5,689,707 to Donnelly.**

13. As to claim 1, Geist teaches a method, comprising in a computer system: receiving a request from a kernel mode driver ("...allocation calls..." Col. 7 Ln. 14 – 29), determining that the kernel mode driver is to be monitored/re-vectoring the request to a kernel mode driver verifier ("...intercept..." Col. 7 Ln. 14 – 29), and taking action in the kernel mode driver verifier to actively test the kernel mode driver for errors ("...value-checking..." Col. 7 Ln. 30 – 36).

Geist is silent with respect to the kernel mode driver verifier being capable of testing the kernel mode driver by simulating a low resource condition including failing requests for memory pool allocation.

Donnelly teaches the kernel mode driver verifier being capable of testing the kernel mode driver by simulating a low resource condition including failing requests for memory pool allocation ("...memory_trigger (expiration event)..." Col. 7 Ln. 24 – 51). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such modification in order to detect memory leaks (Col. 7 Ln. 47 – 51).

14. As to claim 2, Geist as modified in claim 1 teaches the method of claim I wherein receiving a request from a driver includes receiving a function call in a kernel component of an operating system ("...actual memory management functions..." Col. 7 Ln. 40 – 46).

15. As to claim 3, Geist as modified in claim 1 teaches the method of claim 1 wherein determining that the driver is to be monitored includes checking a registry setting ("...list..." Col. 9 Ln. 23 – 27).

16. As to claim 4, Geist as modified in claim 1 teaches the method of claim 1 wherein the request from the driver includes a memory allocation request, and wherein taking action in the kernel mode driver verifier to test the driver includes allocating memory space thereto from a special pool of memory ("...ABLK list..." Col. 7 Ln. 32 – 37, Col. 8 Ln. 27 – 35, Col.. 9 Ln. 64 – 67, Col. 10 Ln. 35 – 63).

17. As to claim 5, Geist as modified in claim 1 teaches the method of claim 1 wherein the request from the driver includes a memory allocation request ("...allocation calls..." Col. 7 Ln. 18 – 21, "...function call..." Col. 8 Ln. 11 – 15).
Geist is silent with respect to taking action in the kernel mode driver verifier to test the driver includes marking memory bounding the memory space to detect improper access of the memory bounding the memory space.
Donnelly teaches taking action in the kernel mode driver verifier to test the driver includes marking memory bounding the memory space to detect improper access of the memory bounding the memory space ("...memory_trigger(expiration event)..." Col. 7 Ln. 24 – 67, Col. 8 Ln. 1 – 9). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such modification in order to detect memory leaks (Col. 7 Ln. 47 – 51).

18. As to claim 6, Geist as modified in claim 1 teaches the method of claim 1 wherein the request from the driver includes a memory de-allocation request ("...removing..." Col. 10 Ln. 45 – 67).

Geist is silent with respect to taking action in the kernel mode driver verifier to test the driver includes marking de-allocated memory space to detect improper access of the de-allocated memory space.

Donnelly teaches taking action in the kernel mode driver verifier to test the driver includes marking de-allocated memory space to detect improper access of the de-allocated memory space ("...free() function..." Col. 8 Ln. 10 – 26). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such modification in order to detect memory leaks (Col. 7 Ln. 47 – 51).

19. As to claim 7, Geist as modified in claim 1 teaches the method of claim 1 wherein taking action in the kernel mode driver verifier to test the driver includes maintaining allocation information in at least one data structure associated with the driver ("...ABLK list..." Col. 7 Ln. 32 – 37, Col. 8 Ln. 27 – 35, Col.. 9 Ln. 64 – 67, Col. 10 Ln. 35 – 63).

20. As to claim 8, Geist as modified in claim 7 teaches the method of claim 7 wherein the request from the driver includes a memory allocation request, and wherein maintaining allocation information includes adding data corresponding to the allocation

request to the data structure ("...ABLK list..." Col. 7 Ln. 32 – 37, Col. 8 Ln. 27 – 35, Col.. 9 Ln. 64 – 67, Col. 10 Ln. 35 – 63).

21. As to claim 9, Geist as modified in claim 7 teaches the method of claim 7 wherein the request from the driver includes a memory de-allocation request ("...removing..." Col. 10 Ln. 45 – 63) and wherein maintaining allocation information includes removing data corresponding to the allocation request from the data structure (ABLK Col. 10 Ln. 45 – 63).

22. As to claim 11, Geist as modified in claim 1 teaches the method of claim 1 wherein taking action in the kernel mode driver verifier to test the driver includes validating call parameters ("...error checking..." Col. 10 Ln. 37 – 44).

23. As to claim 12, Geist as modified in claim 1 teaches the method of claim 1 wherein taking action in the kernel mode driver verifier to test the driver includes checking for resources allocated to the driver at driver unload ("...de-allocation..." Col. 9 Ln. 64 – 67, "...removing..." Col. 10 Ln. 45 – 63).

24. As to claim 13, Geist as modified in claim 1 teaches the method of claim 1 wherein taking action in the kernel mode driver verifier to test the driver includes simulating a low resource condition ("...memory_trigger (expiration event)..." Col. 7 Ln. 24 – 51). It would have been obvious to apply the teaching of Donnelly to the system of

Geist. One would have been motivated to make such modification in order to detect memory leaks (Col. 7 Ln. 47 – 51).

25. As to claim 14, see the rejection of claim 1.

26. As to claim 15, Geist as modified in claim 13 teaches the method of claim 13 wherein simulating the low resource condition includes invalidating driver code and data ("...value-checking..." Col. 7 Ln. 30 – 32).

27. As to claim 16, Geist as modified in claim 1 is silent with respect to the method of claim 1 wherein taking action in the kernel mode driver verifier to test the driver includes checking for timers, in de-allocated pooled memory.
Donnelly teaches the method of claim 1 wherein taking action in the kernel mode driver verifier to test the driver includes checking for timers, in de-allocated pooled memory ("...expiration event..." Col. 5 Ln. 16 – 67, Col. 6 Ln. 8 – 46, Col. 7 Ln. 24 – 51). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 7 Ln. 24 – 51).

28. As to claim 27, Geist teaches a computer-readable medium having computer executable instructions, comprising: receiving a request from a component for allocation of memory space ("...allocation calls..." Col. 7 Ln. 14 – 29, "...function call..." Col. 8 Ln.

11 – 16), determining a location of memory space to allocate ("...ABLK list..." Col. 7 Ln. 32 – 37, Col. 8 Ln. 27 – 35, Col.. 9 Ln. 64 – 67, Col. 10 Ln. 35 – 63) and allocating the memory space (Step 5 Col. 27 – 29, "...allocation is made..." Col. 10 Ln. 45 – 53). Geist is silent with respect to restricting access to areas bounding the location wherein any access request to at least one of the areas results in an access violation and monitoring the areas bounding the location for an access violation.
Donnelly teaches restricting access to areas bounding the location wherein any access request to at least one of the areas results in an access violation and monitoring the areas bounding the location for an access violation ("...memory_trigger (expiration event)..." Col. 7 Ln. 24 – 51). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such modification in order to detect memory leaks (Col. 7 Ln. 47 – 51).

29. As to claim 28, Geist as modified in claim 27 teaches the computer-readable medium of claim 27 having further compute executable instructions, comprising, detecting an access violation ("...MSG list/pool..." Col. 5 Ln. 55 – 67, Col. 7 Ln. 32 – 36, Col. 12 – 22).

30. As to claim 29, Geist as modified in claim 27 teaches the computer-readable medium of claim 27 having further computer-executable instructions, comprising receiving a request from the component for de-allocation of the memory space ("...de-allocation..." Col. 9 Ln. 64 – 67, "...removing..." Col. 10 Ln. 45 – 63).

Geist as modified in claim 27 is silent with respect to restricting access to de-allocated memory space, wherein any access request to the de-allocated memory space results in an access violation and monitoring the de-allocated memory space for an access violation.

Donnelly teaches restricting access to de-allocated memory space, wherein any access request to the de-allocated memory space results in an access violation and monitoring the de-allocated memory space for an access violation ("...track..." Col. 4 Ln. 34 – 67). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 4 Ln. 59 – 63).

31. As to claim 30, claim 29 covers claim 30 since claim 30 is computer-executable instructions of claim 29.

32. As to claim 32, Geist teaches a computer-readable medium having computer executable instructions, comprising: receiving information corresponding to a driver unload ("Routines..." Col. 9 Ln. 64 – 67, "...removing..." Col. 10 Ln. 45 – 63).

Geist is silent with respect to determining whether resources remain associated with the driver; and if resources remain associated with the driver, generating an error.

Donnelly teaches determining whether resources remain associated with the driver; and if resources remain associated with the driver, generating an error ("...memory_trigger(expiration event Col. 7 Ln. 24 – 51: NOTE: Although driver is not

explicitly taught memory allocation is universal to both program and driver). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 7 Ln. 24 – 51).

33. As to claim 33, Geist as modified in claim 32 is silent with respect to the computer-readable medium of claim 32 wherein determining whether resources remain associated with the driver includes examining lists maintained by a system kernel. Donnelly teaches the computer-readable medium of claim 32 wherein determining whether resources remain associated with the driver includes examining lists maintained by a system kernel ("...memory allocation table..." Col. 8 Ln. 1 – 9). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 7 Ln. 24 – 51).

34. As to claim 34, Geist as modified in claim 32 teaches the computer-readable medium of claim 32 wherein determining whether resources remain associated with the driver includes maintaining information tracking memory allocated to the driver and de-allocated thereby ("...ABLK list..." Col. 7 Ln. 32 – 37, Col. 8 Ln. 27 – 35, Col.. 9 Ln. 64 – 67, Col. 10 Ln. 35 – 63).

35. As to claim 39, Geist teaches the system of claim 35 wherein the a request from the driver includes a memory allocation request, and wherein a test by the driver verifier includes allocating memory space thereto from a special pool of memory ("...ABLK list..." Col. 7 Ln. 32 – 37, Col. 8 Ln. 27 – 35, Col.. 9 Ln. 64 – 67, Col. 10 Ln. 35 – 63). Geist is silent with respect to marking memory bounding the memory space to detect improper access of the memory bounding the memory space.
Donnelly teaches marking memory bounding the memory space to detect improper access of the memory bounding the memory space ("...memory_trigger(expiration event)..." Col. 7 Ln. 24 – 51, Col. 8 Ln. 1 – 9). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 7 Ln. 24 – 51).

36. As to claim 40, Geist teaches the method system of claim 35 wherein the request from the driver includes a memory de-allocation request, and wherein a test by the driver verifier includes de-allocating the memory space and marking the de-allocated memory space to detect improper access thereof ("...de-allcoation..." Col. 9 Ln. 64 – 67, "...removing..." Col. 10 Ln. 45 – 67).

37. As to claim 41, Geist teaches the system of claim 35 wherein a test by the driver verifier includes examining resources allocated to the driver ("...searching..." Col. 10 Ln. 45 – 63).

38. As to claim 42, Geist teaches the system of claim 41 wherein examining resources allocated to the driver includes tracking outstanding memory allocated to the driver (ABLK Col. 10 Ln. 45 – 63).

39. As to claim 43, Geist teaches the system of claim 41 wherein examining resources allocated to the driver includes reviewing lists maintained by the operating system component for information therein associated with the driver ("...searching..." Col. 10 Ln. 45 – 63).

40. As to claim 44, Geist teaches the system of claim 35 wherein a test performed by the driver includes validating call parameters ("...error checking..." Col. 10 Ln. 37 – 44).

41. As to claim 45, Geist teaches the method system of claim 35 wherein a test performed by the driver includes failing requests for memory pool allocation ("...(MSG) list..." Col. 7 Ln. 34 – 36).

42. As to claim 46, Geist teaches the method system of claim 35 wherein a test performed by the driver includes invalidating driver code and data ("...value-checking..." Col. 7 Ln. 30 – 32).

43. As to claim 47, Geist is silent with respect to the method system of claim 35 wherein a test performed by the driver includes checking for timers in de-allocated pooled memory.

Donnelly teaches the method system of claim 35 wherein a test performed by the driver includes checking for timers in de-allocated pooled memory ("...expiration event..." Col. 5 Ln. 16 – 67, Col. 6 Ln. 8 – 46, Col. 7 Ln. 24 – 51). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 7 Ln. 24 – 51).

44. As to claim 48, Geist teaches a method in a computer system for verifying system components, comprising: selecting one or more tests for verifying functionality of the system component (Col. 5 Ln. 21 – 42, "...track..." Col. 8 Ln. 11 – 16, "...monitored..." Col. 10 Ln. 1 – 7), modifying a request for system services to include execution of the selected tests/executing the modified request ("...take over..." Col. 5 Ln. 43 – 54, "...intercept..." Col. 7 Ln. 14 – 29, "...thunk..." Col. 7 Ln. 42 – 67).

Geist is silent with respect to one of the tests includes restricting access to a resource such that an attempted access to the resource causes an access violation; and generating errors for any test failures.

Donnelly teaches one of the tests includes restricting access to a resource such that an attempted access to the resource causes an access violation; and generating errors for any test failures ("...memory_trigger(expiration event)..." Col. 7 Ln. 24 – 51, Col. 8 Ln. 1 – 9). It would have been obvious to apply the teaching of Donnelly to the system of

Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 7 Ln. 24 – 51).

45. As to claim 49, Geist as modified in claim 48 teaches the method of claim 48 wherein the system component comprises a device driver (NLM Col. 9 Ln. 9 –13).

46. As to claim 50, Geist as modified in claim 48 teaches the method of claim 48 wherein the request for system services comprises a request to a kernel component ("...kernel...", Col. 9 Ln. 19 – 24).

47. As to claim 51, Geist as modified in claim 48 teaches the method of claim 48 further comprising applying a test condition designed to detect a specific error ("...errors..." Col. 5 Ln. 32 – 42).

48. As to claim 52, Geist as modified in claim 51 teaches the method of claim 51 wherein applying the test condition includes restricting available system resources ("...memory_trigger(expiration event)..." Col. 7 Ln. 24 – 51, Col. 8 Ln. 1 – 9). It would have been obvious to apply the teaching of Donnelly to the system of Geist. One would have been motivated to make such a modification in order to detect memory leak (Col. 7 Ln. 24 – 51).

49. As to claim 53, claim 48 covers claim 53, since claim 53 is a computer-readable medium of claim 48.

50. Claims 10 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,491,808 to Geist, Jr. in view of U.S. Pat. No. 5.689.707 to Donnelly as applied to claims 7 and 37 above, and further in view of PCT WO 95/22104 to Parker et al.

51. As to claim 10, Geist as modified in claim 7 is silent with respect to the method of claim 7 further comprising providing the allocation information to a user interface.
Parker teaches the method of claim 7 further comprising providing the allocation information to a user interface (Step 190 page 20 lines 3 – 34). It would have been obvious to apply the teaching of Parker to the system of Geist. One would have been motivated to make such a modification in order to alert a user of memory allocation problem (page 20 lines 10 – 16).

52. As to claim 38, Geist as modified in claim 37 is silent with respect to the system of claim 37 further comprising a user interface for writing driver information to the registry.
Parker teaches the system of claim 37 further comprising a user interface for writing driver information to the registry ("... SaVE prompt..." page 21 lines 6 – 23). It would have been obvious to apply the teaching of Parker to the system of Geist. One would

have been motivated to make such a modification in order to save user operation (page 21 lines 19 – 23).

Conclusion

53. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat. No. 5,949,972 to Applegate.

U.S. Pat. No. 6,618,824 B1 to Hastings.

U.S. Pat. No. 6,363,467 to Weeks. .

U.S. Pat. No. 5,590,329 to Goodnow II et al.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles E Anya whose telephone number is (703) 305-3411. The examiner can normally be reached on M-F (8:30-6:00) First Friday off.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles E Anya
Examiner
Art Unit 2126

cea

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100